

Steps I took to set up my computer to load interactive Unity-generated VR scenes to an Android phone, and where the steps come from.

1. Downloaded Oculus Mobile SDK (instruction from <https://developer.oculus.com/documentation/mobilesdk/latest/concepts/book-dev-intro/>)
 - a. Steered from there to Unity documentation at <http://docs.unity3d.com/Manual/android-GettingStarted.html>, which says “The best thing to do is follow the instructions step-by-step from the Android Developer portal.”
 - i. At Android Studio and SDK Tools
<http://developer.android.com/sdk/index.html>, downloaded Android Studio, which is only supposed to work up through OS 10.9 and requires Java Runtime Environment 6 and Java Development Kit 7.
 - ii. Using SDK Manager in the toolbar of Android Studio:
 1. Select: Android SDK Tools, Android SDK Platform-tools, and Android SDK Build-tools
 2. “Open the first Android X.X folder (the latest version) and select: SDK Platform, and “a system image for the emulator, such as ARM EABI v7a System Image” (Note, you’ll need to check “Show Package Details” to get to see this option.)
 3. “Click Install X Packages” (Note, this button doesn’t exist, so just clicking “Apply” and it appears to be doing the installation successfully.)
 4. Once complete, click OK and quit Android Studio
 - iii. The first time building a project for Unity you’ll need to add the Android SDK path to Unity if it can’t find the SDK. That can be one in Unity > Preferences, then selecting External Tools. (From docs.unity3d.com/Manual/android-sdksetup.html.)
2. Configure the Android phone for debugging using developer options
 - a. These are available from Home -> All Apps -> Settings -> System -> Developer options. If this is hidden, go to Home -> All Apps -> Settings -> System -> About device; scroll to Build Number; and click on Build Number seven times.
 - b. Once developer options are turned on, enable USB debugging.
3. Install Xcode, Java Development Kit, Android Studio (already covered in 1.a.i. above), and Android Native Development Kit.
 - a. Xcode is available from developer.apple.com/xcode/download
 - b. The Oculus site says Java Development Kit version 8u45 is available from Oracle at <http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>, but it’s actually not. Only a newer version, 8u77, is available there. 8u45 is available in an archive at <http://www.oracle.com/technetwork/java/javase/downloads/java-archive-javase8>

[-2177648.html#jdk-8u45-oth-JPR](#)>. You'll need to make an account with Oracle to download the software.

- c. Android Studio instructions *that aren't quite the same as the ones offered by the Android Studio site* indicate you should download "Android SDK 4.42. API level 19 or later and Android Build Tools 22.0.1". So relaunch Android Studio, check for compatibility with these requirements, and initiate any new downloads if needed. To add Android Build Tools 22.0.1, while in the SDK Tools tab, select "Show Package Details" to select this version, which is otherwise hidden.
- d. Android Native Development Kit (NDK) version r10e has been tested with this Oculus release, but that version is no longer available for download. At the time of writing, only the newest version (r11b) is available.
 - i. The NDK must be extracted to your home/dev folder (~/.dev).
 1. I couldn't find the ~/.dev folder, even when I made all files visible. So the copying couldn't be achieved in Finder. Instead, I ran the following terminal command to copy the downloaded folder from my desktop into ~/.dev:

```
$ cp -R /Users/rz/Desktop/android-ndk-r11b/ ~/.dev/
```
 2. Despite this, and it taking some time to think about copying 700MB (the process it was supposed to be doing), I could not find the android-ndk-r11b folder in /.dev using `ls`. No idea.
4. Follow Samsung VR instructions at <http://www.samsung.com/us/samsungdeveloperconnection/developer-resources/gear-vr/apps-and-games/exercise-1-creating-a-unity-project.html>> to create a first Unity project. While building an empty project for Android, there were several versions of things Unity downloaded though I didn't write down what those things were; I also couldn't find the log at ~/.Library/logs/Unity/Editor.log that the Unity website says I should've found (in fact Unity didn't create any log folder in the Library) so I can't check what it downloaded after the fact.